

**BİLGİSAYAR MİMARİSİ VE BROOKSHEAR
MAKİNESİ**

BLM101

Oğuzhan AKIN

25360859018

1. BÖLÜM: BİLGİSAYAR MİMARİSİ TEMELLERİ

Von Neumann Mimarisi

Modern bilgisayarların çoğu Von Neumann mimarisine dayanır. Bu mimari; verinin ve komutların aynı bellek biriminde tutulmasını öngörür. Temel bileşenler şunlardır:

- **Merkezi İşlem Birimi (CPU)**
- **Ana Bellek (RAM)**
- **Giriş/Çıkış (I/O) Birimleri**

-CPU (Merkezi İşlem Birimi) Bileşenleri-

- **ALU (Aritmetik Mantık Birimi):** İşlemcinin "hesap makinesi"dir. Toplama, çıkarma gibi aritmetik işlemler ile AND, OR, NOT gibi mantıksal karşılaştırmaları yapar.
- **Control Unit (Kontrol Birimi):** Bellekten komutları çeker, kodlarını çözer ve CPU içindeki veri akışını yönetir. Diğer birimlere ne yapacaklarını söyleyen kontrol sinyalleri üretir.
- **Registers (Kaydediciler):** CPU içindeki çok hızlı, geçici depolama alanlarıdır.
- **Program Counter (PC):** Bir sonraki işlenecek komutun bellek adresini tutar.
- **Instruction Register (IR):** O an yürütülen komutu tutar.

2. BÖLÜM: MAKİNE DİLİ VE KOMUT YAPISI

-*Makine Dili (Machine Language) Nedir?*-

İşlemcinin doğrudan yürütebildiği, donanım seviyesindeki tek dildir. Genellikle Hexadecimal (onaltılık) olarak temsil edilse de fiziksel düzeyde 0 ve 1'lerden oluşur.

-*Instruction Set (Komut Seti)*-

Bir işlemcinin tasarım aşamasında belirlenen ve donanımsal olarak desteklediği tüm komutların toplamıdır. Brookshear makinesi, eğitim amaçlı olduğu için sade bir RISC (Reduced Instruction Set Computer) benzeri komut setine sahiptir.

-*Komut Yapısı: Op-code ve Operand*-

Her makine komutu iki kısımdan oluşur:

- **Op-code (İşlem Kodu):** İşlemin tipini belirler (Örn: "Yükle", "Topla", "Atla").
- **Operand (İşlenen):** İşlemin hangi veriler veya adresler üzerinde yapılacağını belirtir.

3. BÖLÜM: BROOKSHEAR MAKİNESİ (APPENDIX C)

Brookshear sanal makinesi, 256 byte bellek (00-FF adresleri arası) ve 16 genel amaçlı register (R0-RF arası) kullanır.

12 Temel Komut ve İşleyışı:

- 1) 1RXY (LOAD):** XY adresindeki hücrenin içeriğini R register'ına yükler.
- 2) 2RXY (LOAD):** XY değerini doğrudan R register'ına yükler (Immediate).
- 3) 3RXY (STORE):** R register'ındaki içeriği XY bellek adresine kopyalar.
- 4) 40RS (MOVE):** R register'ındaki içeriği S register'ına kopyalar.
- 5) 5RST (ADD):** S ve T register'lardaki değerleri toplar, sonucu R'ye yazar (İki tam sayı toplamı).
- 6) 6RST (ADD - Floating Point):** S ve T register'lardaki değerleri kayan noktalı olarak toplar, sonucu R'ye yazar.
- 7) 7RST (OR):** S ve T içeriğine bit düzeyinde OR işlemi uygular, sonucu R'ye yazar.
- 8) 8RST (AND):** S ve T içeriğine bit düzeyinde AND işlemi uygular, sonucu R'ye yazar.
- 9) 9RST (XOR):** S ve T içeriğine bit düzeyinde XOR işlemi uygular, sonucu R'ye yazar.
- 10) AROX (ROTATE):** R register'ındaki bitleri sağa doğru X defa döndürür.
- 11) BRXY (JUMP):** Eğer R register'ındaki değer R0 (Register 0) ile aynıysa, PC'yi XY adresine yönlendirir.
- 12) C000 (HALT):** Programın yürütülmesini durdurur.

4. BÖLÜM: PROGRAMIN YÜRÜTÜLMESİ (MAKİNE DÖNGÜSÜ)

Makine döngüsü, CPU'nun durmaksızın tekrarladığı üç aşamalı bir süreçtir:

FETCH (Getir): CPU, Program Counter'da (PC) yazılı olan adrese gider.

O adresdeki komutu bellekten okur ve Instruction Register'a (IR) yükler.

PC'yi bir sonraki komutun adresini gösterecek şekilde artırır.

DECODE (Çöz): Kontrol birimi IR'deki komutun ilk kısmına (Op-code) bakar.

Hangi işlemin yapılacağını ve hangi register'ların kullanılacağını belirler.

EXECUTE (Yürüt): İlgili birim (ALU veya Veri Yolları) işlemi gerçekleştirir.

Örneğin bir toplama yapılacaksa ALU çalışır, bir veri taşınacaksa bellek iletişimini sağlarıır.